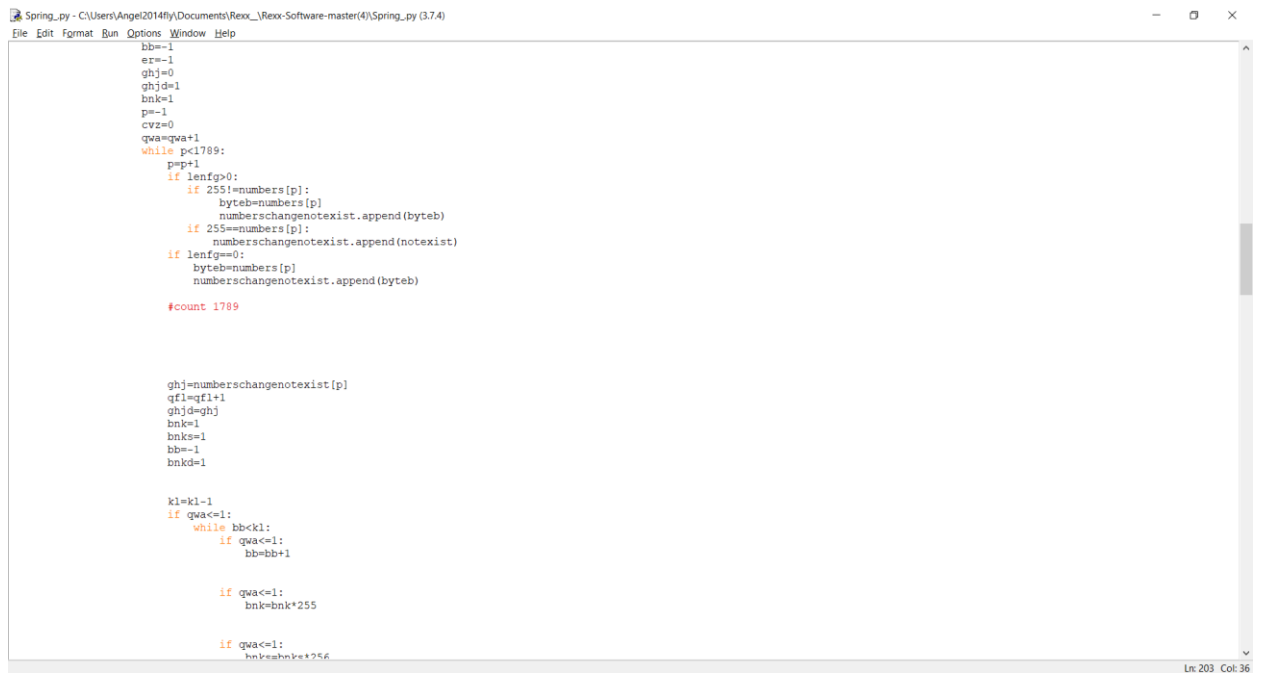


Algorithm Spring made by Jurijus Pacalovas compression by calculus and reverse: Size of file should bigger than 1970 bytes before when want you to compress. if $\text{lenf1} < 2000$:

```
if lenf1<2000:
    print("This file is too small");
    raise SystemExit
```

Size is blocked 1970 bytes. $\text{bnk}=\text{bnk}*255$ $\text{ghjd}=\text{ghj}*\text{bnk}$ $\text{cvz}=\text{cvz}+\text{ghdj}$

$\text{bnk}=\text{bnk}*256$ $\text{ghjd}=\text{ghj}*\text{bnk}$ $\text{cvz}=\text{cvz}+\text{ghdj}$



```
Spring.py - C:\Users\Angel2014fy\Documents\Reix_\Reix-Software-master(4)\Spring.py (3.7.4)
File Edit Format Run Options Window Help
    bb=1
    er=-1
    ghj=0
    ghjd=1
    bnk=1
    p=-1
    cvz=0
    qva=qva+1
    while p<1789:
        p=p+1
        if lenfg>0:
            if 255!=numbers[p]:
                byteb=numbers[p]
                numberschangenotexist.append(byteb)
            if 255==numbers[p]:
                numberschangenotexist.append(notexist)
        if lenfg==0:
            byteb=numbers[p]
            numberschangenotexist.append(byteb)
    #count 1789

    ghj=numberschangenotexist[p]
    qf1=qf1+1
    ghjd=ghj
    bnk=1
    bnks=1
    bb=-1
    bnkd=1

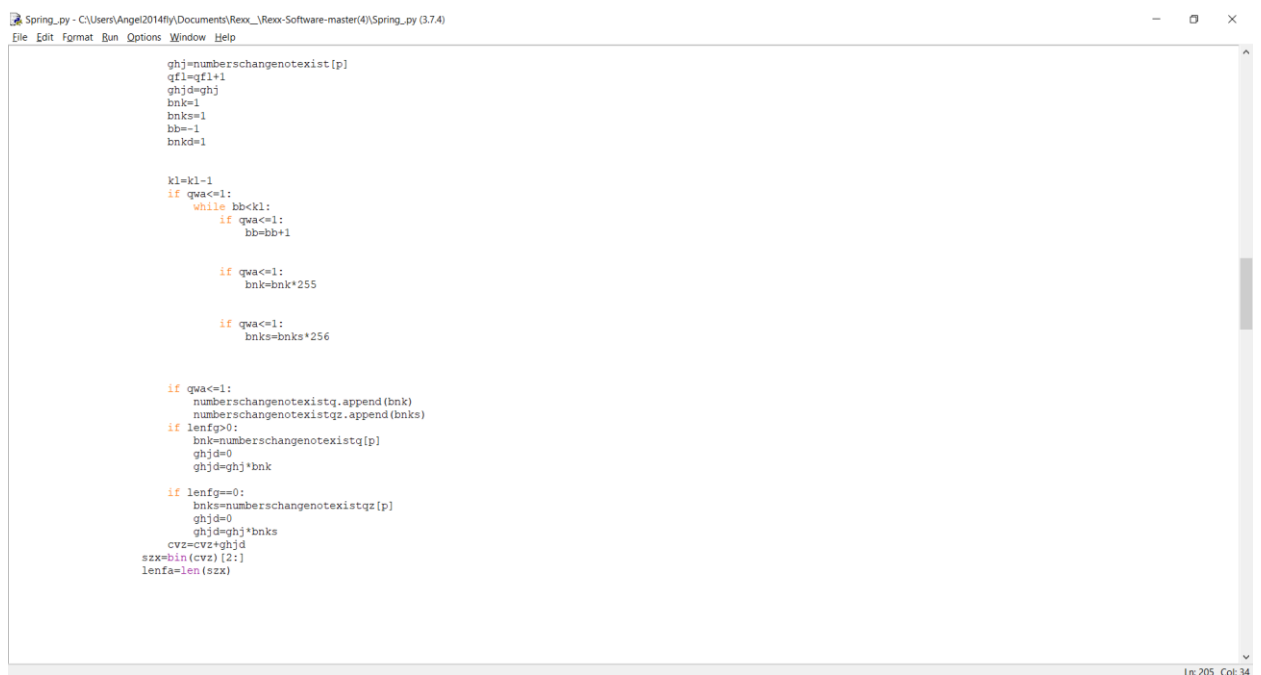
    kl=kl-1
    if qva<=1:
        while bb<kl:
            if qva<=1:
                bb=bb+1

            if qva<=1:
                bnk=bnk*255

            if qva<=1:
                bnks=bnks*256

    if qva<=1:
        bnk=bnk*255
        bnks=bnks*256

    if qva<=1:
        numberschangenotexistq.append(bnk)
        numberschangenotexistqz.append(bnks)
    if lenfg>0:
        bnk=numberschangenotexistq[p]
        ghjd=0
        ghjd=ghj*bnk
    if lenfg==0:
        bnks=numberschangenotexistqz[p]
        ghjd=0
        ghjd=ghj*bnks
    cvz=cvz+ghjd
    szx=b'in(cvz)[2:]
    lenfa=len(szx)
Ln 203 Col: 36
```



```
Spring.py - C:\Users\Angel2014fy\Documents\Reix_\Reix-Software-master(4)\Spring.py (3.7.4)
File Edit Format Run Options Window Help
    ghj=numberschangenotexist[p]
    qf1=qf1+1
    ghjd=ghj
    bnk=1
    bnks=1
    bb=-1
    bnkd=1

    kl=kl-1
    if qva<=1:
        while bb<kl:
            if qva<=1:
                bb=bb+1

            if qva<=1:
                bnk=bnk*255

            if qva<=1:
                bnks=bnks*256

    if qva<=1:
        numberschangenotexistq.append(bnk)
        numberschangenotexistqz.append(bnks)
    if lenfg>0:
        bnk=numberschangenotexistq[p]
        ghjd=0
        ghjd=ghj*bnk
    if lenfg==0:
        bnks=numberschangenotexistqz[p]
        ghjd=0
        ghjd=ghj*bnks
    cvz=cvz+ghjd
    szx=b'in(cvz)[2:]
    lenfa=len(szx)
Ln 205 Col: 34
```

make smaller than size when lenfa<=14305 bits and save size 14307 bits when bigger save size 14320. When size 14320 bits take **0000...111...'0X or 111...0X move to the end 0000...111...0 or 111...0** and change last one to 0. make them together from right to left. Cout this size and save it in bytes than the count of this long and again of this long that should be 1 byte. When sizing 14305 bits save as 1111...0 and size will become 14307 bits.

```

Spring_py - C:\Users\Angel2014fy\Documents\Rexx_\Rexx-Software-master(4)\Spring_py (3.7.4)
File Edit Format Run Options Window Help

    if lenfa>14310 and lenfg==0 and cvb==0 or lenfa<=14310 and lenfg>0 and cvb==0:
        cvb=0
    else:
        j1=data
        if cvb==0:
            f2.write(j1)
            cvb=1

    if lenfa>14305:
        wqve=""
        p=0
        aaqq=""
        d=1
        a=0

        while p<2:
            aaqq=szx[a:d]

            if aaqq=="1":
                a=a+1
                d=d+1
                aaqq=str(aaqq)
                aaqw=aaqw+aaqq
                if aaqq=="0":
                    p=2

            aaqwss=len(aaqw)
            aasqq=""
            ass=0
            asss=0

            ass=aaqwss
            asss=aaqwss-1

            aaad="0"
            aasqq=szx[0:asss]
            aasqq=str(aasqq)
            szx=szx[d:]

            aaqw=""

```

```

Spring_py - C:\Users\Angel2014fy\Documents\Rexx_\Rexx-Software-master(4)\Spring_py (3.7.4)
File Edit Format Run Options Window Help

        aaqq=str(aaqq)
        aaqw=aaqw+aaqq
        if aaqq=="0":
            p=2

        aaqwss=len(aaqw)
        aasqq=""
        ass=0
        asss=0

        ass=aaqwss
        asss=aaqwss-1

        aaad="0"
        aasqq=szx[0:asss]
        aasqq=str(aasqq)
        szx=szx[d:]

        aaqw=""
        zzaax=""
        xc=14320-lenfa
        z=0
        if xc!=14320:
            while z<xc:
                zzaax="0"+zzaax
                z=z+1
            wez=wez+szx
            aaqws=aaqws+zzaax+aasqq+"0"
            szx=""
            zzaax=""
            lenf=len(szx)
            wqve=""
            wqve=szx[0:1]
            if wqve=="1":
                raise SystemExit

    if lenfa<=14305:
        szx=""
        xc=14306-lenfa
        z=0
        if xc!=14306:
            while z<xc:
                szx="1"+szx
                z=z+1

```

```

Spring_py - C:\Users\Angel2014fy\Documents\Rexx_\Rexx-Software-master(4)\Spring_py (3.7.4)
File Edit Format Run Options Window Help
z=0
if xc!=14320:
    while z<xc:
        zaaax="0"+zaaax
        z=z+1
    wer=wer+szx
    aaqws=aaqws+zaaax+aaqq+0
    szx=""
    zaaax=""
    lenf=len(szx)
    wqwe=""
    wqwe=szx[0:1]
    if wqwe=="1":
        raise SystemExit

if lenfa<=14305:
    szx="0"+szx
    xc=14306-lenfa
    z=0
    if xc!=14306:
        while z<xc:
            szx="1"+szx
            z=z+1
        wer=wer+szx
        lenf=len(szx)
        szx=""
        if lenfg>0:
            notexist=k[0]
            szx=bin(notexist)[2:]
            lenf=len(szx)

            xc=8-lenf
            z=0
            while z<xc:
                szx="0"+szx
                z=z+1
            wer=wer+szx
            szx=""

a=0
numberschangenotexist = []
del k[: ]

del numbers[: ]
m = []
h=0

```

```

Spring_py - C:\Users\Angel2014fy\Documents\Rexx_\Rexx-Software-master(4)\Spring_py (3.7.4)
File Edit Format Run Options Window Help

dd=len(aaqws)

szxzzz=""
szxzz=bin(dd)[2:]
dd=len(szxzzz)
xc=8-dd%8
z=0
if xc!=8:
    while z<xc:
        szxzzz="0"+szxzzz
        z=z+1

dd=len(szxzzz)

szxz=bin(dd)[2:]
dd=len(szxz)
xc=8-dd%8
z=0
if xc!=8:
    while z<xc:
        szxz="0"+szxz
        z=z+1

dd=len(szxz)
szxzz=""

szxzz=bin(dd)[2:]
dd=len(szxzz)
xc=8-dd%8
z=0
if xc!=8:
    while z<xc:
        szxzz="0"+szxzz
        z=z+1

wer="0b1"+wer+aaqws+"1"
szx=""

lenf=len(wer)
xc=8-lenf%8
z=0
if xc!=8:
    while z<xc:
        szx="0"+szx
        z=z+1

```

left 0x and 1111....0x

Save how many times was compressed by one byte.

check: if lenfa>14310 and lenfg==0 and cvb==0 or lenfa<=14310 and lenfg>0 and cvb==0:
lenfg mean when the size of data does not exist information on the block and save information
about the first information not exist and change 255 to information not exist. if lenf1<=sssssw or
sssssw<=2000 or qqqwz==255 or cvb==1: check the size of the file and check 255 and cvb mean
check when the size not if lenfa>14310 and lenfg==0 and cvb==0 or lenfa<=14310 and lenfg>0
and cvb==0: and save this file.

